UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/074,030	02/14/2002	John R. Applin	10012812-1	5659

7590   06/15/2005

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER
VU, TUAN A

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 06/15/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<table>
<tr><td rowspan="2">Office Action Summary</td><td>Application No.</td><td>Applicant(s)</td><td></td></tr>
<tr><td>10/074,030</td><td>APPLIN ET AL.</td><td></td></tr>
<tr><td></td><td>Examiner</td><td>Art Unit</td><td></td></tr>
<tr><td></td><td>Tuan A. Vu</td><td>2193</td><td></td></tr>
</table>

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on <u>03 March 2005</u>.

2a) ☒ This action is **FINAL**.   2b) ☐ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) <u>1-21</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) <u>1-21</u> is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1.   This action is responsive to the Applicant's response filed 3/3/2005.

As indicated in Applicant's response, claims 1-4, 7-12, and 15-20 have been amended.

Claims 1-21 are pending in the office action.

Claim Objections

2.   Claims 1, 9, and 18 are objected to because of the following informalities: there appears

to be a missing "the" between 'version of each of' and 'remaining object files' (li. 6, 7, 5, of

respective claims). Appropriate correction is required.

Claim Rejections - 35 USC § 103

3.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person
> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the
> manner in which the invention was made.

4.   Claims 1-16, 18-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Ching, USPN: 6,560,620 (hereinafter Ching) in view of Leblang et al., USPN: 5,649,200 (

hereinafter Leblang).

As per claim 1, Ching discloses a method for verifying a version for each of a plurality

of files in a computer program, the method comprising steps of:

identifying a version of a selected file of the plurality of files included in the computer

program (col. 7, line 64 to col. 8, line 15);

comparing version of said selected file with version of each of the remaining files of the

plurality of file (e.g. *with any selected second list* - col. 2, lines 40-42; col. 8, lines 45-49; col. 9,

Fig. 7; line 61 to col. 10, line 24 – Note: Versioned sections of document being separately stored

reads on plurality of versioned files – it is more meaningful to store a different content in

separate file version than under a same version file -- being compared with the selected file, and

category of same type being listed reads on comparing one selected file against plurality of same

category of versioned files); and

generating an alert in response to the version of said selected file being different than the

one or more versions of the remaining files (e.g. *display on the user computer* - col. 2, lines 18-

42).

But Ching does not explicitly disclose that the file being selected for comparing is a

object file; however, Ching discloses a wide possibility to apply the comparing tool aiming at

various the file content like programming languages (*software language ... arbitrary* - col. 15,

lines 22-38). Leblang, in a tool to merge different version file being stored analogous to Ching,

disclose object files (col. 1- 2; *msg.o, foo.o, bar.o* – Fig. 20, 24). In view of the same endeavor

for detecting file differential by both Leblang and Ching, it would have been obvious for one of

ordinary skill in the art at the time the invention was made to make the file type by Ching's tool

so that it be also a object file as taught by Leblang because file as persisted can be reused in

software application rebuilding (see Leblang: col. 1 line 66 to col. 2, line 42) and according to

Ching's approach to alleviate application user effort (see col. 2), any file type that can be stored

in computer (col. 7, lines 12-18) can be used in this comparing tool to alleviate extensive

resource.

As per claim 2, Ching discloses a tool/kit to effect file comparison and Leblang discloses

developers with merging, release building with auditing tools (see Leblang: Fig. 1, 6-7, 23); and

in view of the rationale as set forth in claim 1 combining object file by Leblang, Ching discloses

plurality of files includes a respective version related to software developer's kits (Fig. 7, 9 –

Note: the software developer's kit for comparing object file has been rendered obvious with

Leblang).

As per claim 3, in light of the rejection in claim 2, Ching identifying a version of an

object file (Note: when the list of versioned file is displayed by hierarchy of version number,

identifying version from a object file version is self-evident - Fig. 7, 9), storing the version of the

selected file as a variable (Note: since this comparing tool is based on database, the persisting of

file name therein for retrievel via code tool inherently teaches version defined in variable)

As per claim 4, in light of the rejection in claim 3, Ching the step of storing the list of

object files presented for comparison as versions defined as variables used in the application

code.

As per claim 5, Ching discloses generating an alert informing the user of a version

mismatch (e.g. *display on the user computer* - col. 2, lines 18-42).

As per claim 6, see time stamp (col. 10, lines 18-21).

As per claim 7, Ching discloses displaying a respective textual message for the object

file in response to the selected file being different than the version of one or more object files (

see col. Fig. 5-8).

As per claim 8, Ching discloses storing the respective textual message of one of the

plurality of files as an initial textual message in response determining the initial textual message

equals a null value (Note: persisting file version with initial text belonging to very first, e.g.

version 0 according to well-known practice - in memory/database to keep track of version is

disclosed in Ching's teaching – see Fig. 2, Fig. 6, 7, 9); and displaying the initial textual

message in response to the identified version being different than the initial version (refer to

rejection of claim 7).

As per claim 9, Ching discloses a computer readable medium on which is embedded a

program, the program performing a method for verifying a version for each of a plurality of files

in a computer program, the method comprising steps of:

identifying (selected … file … plurality of files);

comparing (version …selected … file … remaining … plurality of files); and

generating (an alert … version … selected … being different remaining … plurality of

files); all of which steps having being addressed in claim 1.

But Ching does not explicitly teach that the selected file is an object file among a

plurality of object files stored in a computer program. However, the object file being versioned

has been addressed in the rationale as set forth in claim 1 using Leblang.

As per claims 10-16, these are computer medium claims corresponding to claims 2-8;

hence are rejected with the rejections as set forth therein, respectively.

As per claim 18, Ching discloses an apparatus comprising means for:

identifying (selected … file … plurality of files);

comparing (version …selected … file … remaining … plurality of files); and

generating (an alert … version … selected … being different remaining … plurality of

files); all of which steps having being addressed in claim 1.

But Ching does not explicitly teach that the selected file is an object file among a

plurality of object files stored in a computer program. However, the object file being versioned

has been addressed in the rationale as set forth in claim 1 using Leblang.

As per claims 19-21, these are apparatus version claims corresponding to claims 2, 5,

and 6; hence are rejected with the rejections as set forth therein, respectively.

5.   Claims 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bowman-

Amuah, USPubN: 2001/0052108 (hereinafter Bowman) in view of Ching, USPN: 6,560,620.

As per claim 17, Bowman discloses a software development kit configured to facilitate a

development of an application comprising a plurality of files, the software development kit

comprises:

a plurality of modules (pg. 16, para 0471; para 0040; para 1172-1175 – Note: object

oriented object or module and version tool reads on object file being stored); a time stamp being

utilized to identify plurality of file such as incident reports (*time stamps* -- para 2165-1272);

wherein the application developed with the software development kit is configured to generate

response when detecting of a conflict caused by two of the files having different version (e.g.

version control – para 1703; para 1172-1175, para 1180-1185 – Note: version control tool used

by developers to synchronize time recorded versions of earlier in the version database reads on

generating a message informing the developer of such version difference in order to prevent

conflicts – hence a error message -- in using non-synchronized version).

But Bowman does not explicitly disclose time stamp to identify corresponding version of

software kit for each plurality of object files; nor does Bowman disclose module configured to

embed the time stamp within at least one of the plurality of files. It was well-known that version

control for recording and tracking changes to files in a software development management

environment is time related and even Bowman discloses this (see para 1175). Hence, having a

time stamp to differentiate version of software file stored within version control repository is

strongly implied or would have been obvious in view of such suggestion. Ching's tool for

synchronizing software file discloses time stamp (col. 10, lines 13-23). In view of the endeavor

to avoid conflicts as shown by Bowman, it would have been obvious for one of ordinary skill in

the art at the time the invention was made to apply a time stamp to each of the versioned files in

the Version Control database by Bowman because time-based parameter like time-stamp is

known to be a differentiating attribute used in persisting data, files in many applications of the

likes of Bowman's version control tool and Ching's repository of versioned files of a same

category thus used in well-known version identifying endeavors.

Response to Arguments

6.   Applicant's arguments filed 3/3/2005 have been fully considered but they are not

persuasive. Following are the Examiner's response in regard thereto.

Claims rejection under 35 USC §103:

(A)   Applicants have submitted that Leblang does not teach object file and Ching teaches

comparing contents of documents (Appl. Rmrks pg. 8, last 3 para). Ching teaches a tool to

compare versioned files and in the course of such process by selecting a specific version, enable

more of related versioned file instances sharing a same category relationship to be listed and

enable their comparison wherein file contents are sequentially matched. The process of

comparing contents of versioned documents or files is integral to every version control tool; and

because the claim does not explicitly enforce the understanding that by comparing versions, there

will be absolutely no going into the content of the versioned files being compared, the argument about content comparing is alleged or misplaced. Ching's tool enable selection of file to compare and detect discrepancies during the comparing process with related versions of that file; and this reads on 'comparing the version of selected ... with version ... plurality ... files' as recited in the claim. The fact that Ching mentions that any file format or type can be used as file for comparison has lead to the version control, merging and auditing disclosed by Leblang. The common endeavor is to have persisted versions of files that can be reused for subsequent application so that systematic rebuild of files without comparison would not be obviated thus making the reuse effort less extensive. Both references is using version comparing so that only the differences of the files being compared are exhibited to the user/developer using the selected file for some targeted application, e.g. a browser application for Ching, or a release enlistment of software files by Leblang. The rejection has set clear why the type of file to persist can be more than text file type and why that would have been obvious. The Applicants appear to have failed to show why making the versioned documents by Ching so that they are object files would not be appropriate. In response to applicant's argument that Leblang's file is not object file, the rejection has shown example of object being assembled to clarify this issue. Applicants contend by attacking each prior art reference piecemeal in disregard to why the references have been put together. In response to this, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention particularly when there is some teaching, suggestion, or motivation to do so found either in the references themselves, or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347,

21 USPQ2d 1941 (Fed. Cir. 1992). In this case, Ching has provided enough suggestion or teaching for rendering the use of Leblang appropriate. The burden for Applicants is to show exactly how such combination would be prone to generating adverse effects or lead to negative teachings.

(B)   Applicants have submitted that combining Leblang and Ching would not result in 'verifying a version … in a computer program … generating an alert … within the computer program' (Appl. Rmrks pg. 9, top para). The rejection has provided teachings from Ching to 'meet verifying a version … generating a alert…' with the only exception of the nature of the file under version comparison process. And the motivation to make this type an object file as shown in the software assembling tool with merging tool by Leblang as an obvious motivation based on Ching's teaching has been put forward. The Applicants seem to allege that the prior does not meet a claimed limitation without pointing out how the claimed invention specifically distinguishes the prior art as applied. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

(C)   Applicants have submitted that Bowman merely teaches time stamps for assist logging and inquiries processing and that Ching or Bowman does not have object file (Appl. Rmrks pg. 9, last 2 para, pg. 10, top). The rejection has pointed out how Bowman's version control tool reading on storing file or object-oriented class -- or reusable object --files used in developing code; and furthermore has cited Bowman's portions showing version differences and possible conflicts in synchronized concurrent usage of versioned copies of files by plurality of

developers; and has explained the self-evident teaching based thereon that upon detection of files

version conflict a message to that effect is evident because a version tool is fundamentally a

interactive tool for user to be notified of such mismatch. And further, the rejection has also

explained how time stamp is recognized as an attribute essential in establishing a unique

characteristic used in differentiating time-based versions, with evidence in part from Bowman

and also from Ching. Therefore, the arguments amount to mere allegation without pointing

exactly to where the combination as set forth would be inappropriate, in view of the suggestion

or teachings perceived by one skill in the art when exposed to Bowman's version control tool

and Ching's time stamp usage.

For the above reasons, the claims stand rejected as set forth in the Office Action.

Conclusion

7.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time

policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing

date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner

should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can

normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is

assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before

using) or 703-872-9306 (for official correspondence) or redirected to customer service at 571-

272-3609.

Any inquiry of a general nature or relating to the status of this application should be

directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
June 8, 2005

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100